DOCUMENT RESUME

ED 319 372 IR 014 390

AUTHOR Kurland, D. Midian

TITLE Software in the Classroom: Issues in the Design of Effective Software Tools. Technical Report No. 15.

INSTITUTION Bank Street Coll. of Education, New York, NY. Center for Children and Technology.

PUB DATE Apr 83

NOTE 13p.; Paper presented at the Annual Meeting of the American Educational Research Association (Montreal, Quebec, Canada, April 11-14, 1983).

PUB TYPE Viewpoints (120) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC01 Plus Postage.

DESCRIPTORS *Computer Assisted Instruction; *Computer Software; *Computer Software Development; Elementary Secondary Education; *Microcomputers; Models

ABSTRACT

This paper identifies three ways that computers are used in educational contexts. The first and most widespread use is as a tutor, i.e., as a delivery system for programmed instruction and drill-and-practice activities. The second use is as a programming environment to teach programming languages such as BASIC, LOGO, or PASCAL. The third use is as a flexible, reconfigurable tool, employing software that can turn the computer into a word processor, calculator, music system, data organizer, graphing system, note taker, or bulletin board. The first two models (computer as tutor and as programming environment) are discussed briefly, and the third mode (computer as tool) is discussed in more detail. Various types of software and applications in this mode are described. It is argued that, although this is the most neglected of the three approaches, it is the type of use which holds the most promise for computer use in the classroom. (MES)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED319342

CENTER FOR CHILDREN AND TECHNOLOGY

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
X This document has been reproduced as received from the person or organization originating it.
☐ Minor changes have been made to improve reproduction quality.
• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY
Katherine McMillan
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

IR014390

Software in the Classroom: Issues in the Design of Effective Software Tools

D. Midian Kurland

Technical Report No. 15

Bank Street College of Education 610 West 112th Street NY, NY 10025

BEST COPY AVAILABLE

Software in the Classroom: Issues in the Design of Effective Software Tools

D. Midian Kurland

Technical Report No. 15

SOFTWARE FOR THE CLASSROOM: ISSUES IN THE DESIGN OF EFFECTIVE SOFTWARE TOOLS*

D. Midian Kurland

Center for Children and Technology
Bank Street College of Education

According to current estimates, by the end of 1983 more than 1,000,000 microcomputers will be in place in classrooms across the country. Spurred by lowering costs as well as by pressure from parents and educational policy makers, schools are purchasing computer hardware at an accelerating rate. As the price of peripheral devices such as disk drives and printers also falls, it is becoming possible to envision a time in which every classroom will have its own computer system. While the availability of inexpensive and reliable computer hardware is fast becoming a reality, the future of microcomputer technology in classrooms hinges on appropriately designed application software, and on how teachers are able to incorporate it into their classrooms. How well or how poorly available software helps teachers to accomplish their classroom goals will determine whether computers will be the powerful instructional force which its proponents foresee, or merely another educational innovation that becomes nullified through partial assimilation by the educational system.

How well, then, is educational software keeping pace with the rapid increase in available hardware? Is there software appropriate for the many uses to which a computer can be put that is geared to the user environments of schools? In general, the answer to both these questions is no. If one takes a look at any of the educational computing magazines or talks to teachers struggling with their first computer, a general impression emerges that educational software is often poorly conceived, "buggy," difficult to use, difficult to integrate with the rest of the curriculum, and designed without sufficient regard for the range of needs and abilities of students. While it would be unfair to

---

*Paper presented at symposium American Educational Research Association meetings, "Chameleon in the Classroom: Developing Roles for Computers," Montreal, Canada, April 1983.

the authors of the few interesting and well-executed educational programs to brand all educational software as inadequate, the fact remains that the vast majority of software produced for schools is amateurish, unimaginative, or both.

What should a teacher or administrator do about software when the first computers roll through the door? How can nonspecialists make sense of the mountain of available programs? How can they select programs of high quality which fit with the educational philosophy and instructional orientation of their schools? How can software help those teachers who are new to computers better understand their potential applications? Clearly, before teachers and administrators can make intelligent decisions about what software to purchase for their computers, they must have clearer ways of thinking about what a computer is for, and how different types of software reflect these different purposes.

There are at present three ways in which computers are being used in educational contexts--as conveyers of content, as programming environments, and as problem-orientated tools. Implicit in each use is a different model of what a computer is and what its role in the classroom should be. I will briefly discuss the first two models--computer as tutor and computer as programming environment; then I will discuss in more detail the third model--computer as tool. Though currently the most neglected of the three approaches, I will argue that it is this type of use which holds the most promise for computer use in the classroom.

Computer as Tutor

One of the earliest uses of computers in the classroom, and still the most widespread, is as a delivery system for programmed instruction and drill-and-practice activities. The use of computers to deliver "courseware" has been promoted by many of the large educational publishers who want software that supports and looks very much like the textbooks they produce. Such uses of the computer are also promoted by most educational computing magazines and conferences through what they choose to focus their reviews and critiques on. When looking for classroom software, teachers frequently first turn to the published software reviews that are provided by EPIE, Classroom Computer News, Electronic Learning, and other computer-oriented periodicals. These reviews focus almost exclusively on "instructional software" in the form of drill-and-practice programs and/or games. Putting aside the question of how to evaluate software--how to decide which features of different types of programs to evaluate; how to weight pluses and minuses of individual features in arriving at an

overall rating; and how to maintain the consistency, quality, and reliability of the reviewers--by following this path to acquiring software, the school inevitably ends up with a large number of individual, unrelated programs, each aimed at some specific skill or concept. Since approximately 90% of the programs reviewed for the educational market are based on instructional drill-and-practice formats, these review efforts tacitly support using the computer as the transmitter of predefined content, and placing the student in the role of receiver or respondent.

Computer as Programming Environment

Using the computer to run packaged courseware, regardless of its quality, has many limitations. While it does give the classroom computer something to do, in many cases it is not clear that using the computer in this way has any real advantage over older instructional technologies such as flashcards and workbooks. In contrast, there is a growing contingent of educators who choose to bypass the issue of what software to use by having their students focus all their efforts on learning a programming language such as BASIC, Logo, or PASCAL. This approach has the advantage of being inexpensive, since most computers come with one or more languages installed, and students of all ages can, in theory, be taught the same language using the same software. Further, knowing how to program a computer is a valued skill outside of the school, whereas knowing how to sit through a drill-and-practice lesson is not. Finally, learning to program may teach the student something about logic, reasoning, and problem solving that can potentially be applied in domains far removed from programming.

Using the computer to teach about computers and programming does have its merits. However, this approach provides students and teachers with a rather narrow view of what a computer is for and what it can do. Further, programming, especially for novices, is a difficult activity to put into a context that makes clear the goals and purpose of learning to program. Programming a computer is difficult work. It takes a great deal of time and effort before a student is capable of using his or her programming skills in the service of real problems in other curriculum areas. Commercial software programs exist precisely so that one does not have to spend hours or days writing a program each time there is a problem to be solved. It is unrealistic to deny students the use of commercial software and, at the same time, expect them to use the computer to assist them in their coursework. Thus, unless the teacher has the time and is highly skilled at preparing programming "microworlds," programming tends to remain a separate subject that must be taught in addition to

the rest of the curriculum. This puts a significant strain on already overburdened teachers who may not even have enough time for teaching the traditional subjects such as science and social studies.

Opting for the programming route also runs into the problem that students are introduced to programming in a manner which bears little resemblance to the way programs are actually written by professional programmers. No experienced programmer would ordinarily write a program without a software library of programming utilities and aids. A programming language is merely one part of a complete programming environment that should include program editors, debugging utilities, simulators, graphics and music editing systems, and special input and documenting routines to speed up coding. Using the computer to teach programming does not eliminate the need for software; it simply changes the type of software needed.

Computer as Tool

An alternative way of conceptualizing the role of the computer in the classroom is to think of it as a flexible, reconfigurable tool. A tool, by definition, is anything that a person can use to get a job done faster, better, and/or with less effort. Schools are full of specialized tools to help the student through the curriculum. Math tools such as the protractor, ruler, and compass help the student to acquire the concepts of geometry and measurement. Similarly, pencils, erasers, scissors, and tape are tools that assist the student to produce and revise texts. Just as these traditional instructional tools can be used in the service of a diversity of curriculum goals, the computer equipped with tool software can serve many purposes. The class of tool software consists of programs which turn the computer into a word processor, calculator, music system, data organizer, graphing system, note taker, or bulletin board. None of these types of programs teaches a specific content, but neither are they to be used purely as an adjunct to the regular curriculum. By serving established needs, such as creating and revising a composition or producing a graph of a simple function, tool software is easily understood and assimilated by teachers and students regardless of their prior level of experience.

Rather than replacing the teacher--the implicit goal of the computer-as-tutor model--the computer-as-tool model puts the teacher back into the thick of the educational process, armed with an enlarged array of powerful instructional tools. However, while the idea of using the computer as a tool for many different purposes is appealing on a number of levels, the quality of available software tools poses a serious problem for the educator. Put bluntly, there are almost no

software tools which have been produced for the educational market, and what there are tend to be poorly thought out and difficult to use. However, many extremely powerful and well-executed software tools have been created for the business world. While educational publishers are churning out programs that look like workbook pages, business programmers are producing powerful, yet easy-to-use tools. These software tools not only facilitate traditional business activities, but make possible whole new types of activities that, previously, were either too time consuming and/or too difficult to be practical.

Consider, for example, the VisiCalc program. Throughout the sixties and early seventies, computerized business systems replicated functions done by older technologies (payrolls, inventories, mailing lists, etc.). Similarly, most current educational software replicates the type of drill-and-practice activities previously done with flashcards or worksheets. The early users of business systems had little experience with the capabilities of the technology; hence, they demanded nothing more innovative.

However, when VisiCalc was introduced in the late seventies, it began to change the way the business world thought about the computer. Designed as a general-purpose accounting program, VisiCalc was easy to use, even by those with no previous computer experience. More importantly, it facilitated the manipulation of information in ways that were novel to users. They could now model a wide variety of situations by systematically varying the design of the parameters. The program automatically calculated and displayed the effect of changing one parameter on the remaining variables in the model. This piece of software provided new means for thinking about and manipulating information; simultaneously, it made users aware of the novel functions of the computer.

VisiCalc has been followed by a whole generation of innovative office software. Relational data-base management systems, sophisticated word processors, and electronic mail and other communication systems are examples of programs which, over the past few years, have radically altered information use and management in the business world.

The VisiCalc example yields an important lesson about the interaction of software and context. The software was initially recognizable and usable as something that fit into the work needs of managers and executives. Yet, the experience of exploring and exploiting the potential of the software yielded something much more powerful than numbers arranged neatly on the page--namely, new ways of thinking about business information.

Such developments should also be possible in the educational realm. However, until producers of educational software begin to produce more software tools, it will be necessary for educators to seriously consider products developed for other markets and types of users. For the present, an interesting problem facing educators is how to adapt the professionally developed products created for the business market to the needs of teachers.

What is needed is software which teachers can immediately see as helping to do the current work of the classroom, but which also supports new ways of working, learning, and teaching. An important goal for software being produced for today's classroom is that it teach the child, but also illustrate the open-ended potential of this technology for teachers.

A word processor is a wonderful example of just such a program. It does not introduce a new activity into the classroom. Writing is simply accomplished through a different medium. Studies under way in our laboratory indicate that with the right software, students can start writing on the computer the day it comes into the room. Stories that were started on paper can be moved onto the word processor with little trouble. The teacher need not invent new activities, new projects, or take time to teach a new subject before students can use the computer.

Compared to the many other potential uses of the classroom computer, word processing is relatively easy for noncomputer-experienced teachers to try. However, while using a word processor may initially seem like simply substituting the keyboard for the pen, once students start using the word processor, unanticipated effects begin to appear. For example, in a recent study, a teacher reported that when the computers came into her classroom, she found herself immediately starting to correct the students' papers in ink instead of pencil. Asking them to revise their work no longer required a major production effort from the students. Marking up a text did not mean requiring the student to recopy it. This immediately freed the teacher to request more specific revisions from the students. Students reported, and their writing results appeared to indicate, that they wrote longer pieces and revised their writing more often when they could use the word processor. There also was a greater tendency to collaborate on a piece of writing and to read each other's writing since to do so was now much easier with the availability of typed copy and text displayed on a large screen.

The teacher discovered that conferences at the computer with the student's text on the screen was a wonderful way to display, discuss,

and revise a piece of writing. Changes could be made as they were discussed. At the end of the conference, the student had a typed, revised copy instead of a marked-up piece of paper to be recopied. Thus, working with a word processor began to change how the writing process took place in the classroom on a number of levels, in addition to facilitating the mechanical aspects of producing a text.

Besides creating new ways for teachers and students to think about certain activities, software tools can also be used in more traditional ways when appropriate. Use of a word processor need not be limited simply to composition. By preparing paragraphs, sentences, or word lists and then storing them in files on a disk, a teacher can create grammar, spelling, and style exercises tailored to the needs of individual students or to the whole class. This type of tool software can accomplish more effectively what a stack of drill-and-practice language arts programs are required to do, while at the same time acquainting the student with a tool that can be applied to a wide range of problems in the future.

Word processors are good examples of software tools which, though originally developed for the office environment, can be and have been brought into the classroom and utilized effectively. However, software designed for one environment is not always well suited for another. In the case of word processors, many have been designed around the needs of secretaries. These programs are intended to be used on a daily basis to enter, revise, and print someone else's text using powerful but complex editing and formatting commands. Thus, these word processors are often unnecessarily complicated--difficult to learn or to remember how to use. Similarly, systems for office use may include functions that, while useful in the office, may not be educationally sound in the classroom. Consider the example of programs which can catch spelling errors in documents.

The current state of the art allows the user of a microcomputer to check a document against a 20- to 50-thousand word dictionary stored on a floppy disk. The program can mark each word in the document that is not in its dictionary, suggest a more likely spelling, and make the correction with or without the student's assistance. In addition, such programs can evaluate grammar, word usage, sentence structure, tone, and a number of other stylistic parameters which might be of interest to the writer. Because currently available computer software can do such tasks for the writer, educators are forced to think hard about what it is we want tool software to do. Will the use of these spelling checkers seriously lower the spelling ability of students? If it does, does this really matter any more since we have spelling checkers? Or should schools insist on spelling checkers

which simply report how many suspicious words are in a text or in a paragraph and leave it to the student to try to find them? Or should the dictionary be available only for students to look up specific words that they are unsure of as they write?

In light of the almost total lack of research into such issues, it seems that the best option at this point is for software designers to implement each of these levels of spelling checking, and let the individual teacher set the program for the type of assistance the students can have. In this way, the teacher could make available an on-line dictionary while the student is writing, and let the student use the automated proofreading facilities only after final drafts of a piece of writing have been accepted.

Today, if a teacher uses a spelling program in the classroom, he or she has little control over the form of assistance it provides. Nonetheless, until spelling checkers are developed that are sensitive to instructional needs, the currently available spelling checkers, if used creatively, can be effective in helping students with their writing.

Word processors and spelling checkers are examples of tools which primarily facilitate the production end of the writing process. Different types of existing software tools can be used to facilitate other aspects of writing. Data-base management systems, for example, can be employed effectively as prewriting tools to help the student collect, organize, and discover patterns and relationships in information before beginning a report. Simple to learn and use, data-base systems have existed for the popular microcomputer for the past several years. They can be used collectively by a class to enter information on some topic--for example, marine animals--after which students, individually or in small groups, can interrogate the data base as a means of exploring such questions as: What mammals live in the ocean? Do carnivores tend to live close to the shore? What kinds of sea animals have commercial value? and so on. Reports generated by data-base systems can then be used as notes for the students when they write up their findings.

Tools such as data-base management systems do not teach about a subject, nor do they directly lead the student into entering the best information in the best way or in framing the best searches for that information. They simply make it more possible for that to happen. However, not all writing tools need to be this open-ended. For example, the Planner, one of the Quill Writing Programs being developed at Bolt Beranek and Newman, Inc., helps students to consider the types of information to include when doing a specific type of writing. Planner facilitates the precomposing stage of the writing

process by helping the student decide what content to include in his or her writing through a process of electronic brainstorming. The program is designed with a series of templates which can be modified or added to by the teacher or students. Each template is used for a different type of writing or subject area, such as a movie review, persuasive essay, or newspaper story. The student selects one of the Planner templates, and then is asked a series of pertinent questions about the selected topic. For example, the teacher can set up a Planner template for marine animal reports. When the student uses the Planner program, a series of questions appears on the screen which might request that the student type in the topic, the animal to be discussed, two of its characterizing features, two ways in which it is of importance to man, what it likes to eat, where it lives, and what is special about it. When the student is finished with the Planner, a copy of the prompts and the student's answers is printed out as an outline to be used when writing the report.

Conclusion

While the focus of this paper has been on tools to facilitate the writing process, powerful software tools that put the user in control of the computer exist for many purposes. They are available for music, graphics, math, communications, and many other domains. Unlike most of the software that is finding its way into schools, tool software represents an open-ended potential limited only by the imagination of the teachers and students who use it. A program such as Spelling Demons, which drills students on the 100 most frequently misspelled words, can never do any more than that. However, a word processor for approximately the same price can be used not only for spelling lessons, but for other activities as well by children of most grade levels and abilities.

The greatest strength of the computer lies in the fact that it can be quickly reconfigured to be many different kinds of tools--word processor, graphics editor, data collector, music composition aid, calculator, and proofreader, as well as tutor or tester. When considering what software to use with the classroom computer, the issue is not simply what content area the software addresses, but how the content is approached, how much control the student has, and how wide a range of uses the program serves. In order to improve and better understand the use of software in classrooms, research is needed to analyze the current software tools being used in schools; to evaluate the many other software tools which are currently available, irrespective of their intended audience; and to develop better means of integrating software tools into the classroom context. As our society moves into the information age in which access to information will be

mediated more and more through computers, teaching students to use software tools effectively will become an increasingly important role for schools. It is time for both educators and software producers to recognize this, and to begin a serious effort to develop the tools that will provide access to this rapidly approaching world.